第29卷 第2期 计算机工程与设计 2008年1月
Vol.29 No.2 Computer Engineering and Design Jan. 2008

扰乱认知能力的粒子群算法

冯玉宇， 周育人
(华南理工大学 计算机科学与工程学院，广东 广州 510640)

摘 要：受遗传算法“杂交”思想的启发，提出一个改进带收缩因子的粒子群算法。该算法在带收缩因子的粒子群算法中增加扰乱粒子认知能力的方法。即对粒子i，随机选择另外一个粒子j，按照一定的概率用粒子j的当前位置替换粒子i的当前位置。为了检验新算法的性能，选用5个高维函数进行了测试，实验结果表明，改进的算法不仅具有良好的稳健性，而且还有良好的收敛性。

关键词：粒子群算法；认知能力；扰乱；稳健性；收敛性
中图法分类号：TP18 **文献标识码**：A **文章编号**：1000-7024 (2008) 02-0401-04

Particle swarm optimization of disturbing cognitive capability of particle

FENG Yu-yu, ZHOU Yu-ren
(College of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China)

Abstract: Enlightened by the crossover of genetic algorithm, a novel algorithm of improving particle swarm optimization with constriction factor by disturbing the cognitive capability of the particle is presented. For each particle i in the swarm, another particle j is selected. The current position of the i-th particle is replaced by the current position of the j-th particle randomly. Five high-dimensional functions are selected to test the performance of the new algorithm, the result of experiment verified that the proposed algorithm had better stabilization and convergence.

Key words: particle swarm optimization; cognitive capability; disturbing; stabilization; convergence

0 引 言

粒子群算法(particle swarm optimization, PSO)是由Eberhart和Kennedy等人于1995年提出的一类群智能随机优化算法，它源于对鸟类觅食行为的模拟。由于PSO算法思想简单、实现容易、需要控制的参数少而被广泛应用于图像处理、模式识别、神经网络训练等领域。

PSO算法与其它进化算法一样，存在早熟收敛问题。为了避免早熟收敛，一些研究人员提出增加种群多样性的思想，现在主要有两类：

(1)强制改变粒子搜索路径的算法，如：模糊适应的PSO (fuzzy adaptive particle swarm optimization)[1]、自适应环境的PSO (adapting particle swarm optimization to dynamic environments)[2]等；

(2)与其它进化算法相结合提出的混合粒子群算法。本文正是受遗传算法“杂交”思想的启发而提出的混合粒子群算法。

1 相关的PSO算法

1.1 惯性权重线性递减的PSO——PSO-LIR

假设在D维搜索空间中，种群规模为n，第i个粒子的位置和速度分别是$\boldsymbol{X}_i=(x_{i1},x_{i2},\cdots,x_{iD})$和$\boldsymbol{V}_i=(v_{i1},v_{i2},\cdots,v_{iD})$，$i=1,2,\cdots,n$，第i个粒子搜索到的最好位置是$\boldsymbol{P}_i=(p_{i1},p_{i2},\cdots,p_{iD})$，整个种群搜索到的最好位置是$\boldsymbol{P}_g=(p_{g1},p_{g2},\cdots,p_{gD})$。对于一个全局优化问题Min$\{f(\boldsymbol{X})\}$，将粒子的位置$\boldsymbol{X}$带入函数$f(\boldsymbol{X})$计算其适应值，适应值的大小代表粒子位置的好坏。

在粒子群算法中，第i个粒子的第d维速度和位置更新方程按式(1)和式(2)计算[3]

$$\boldsymbol{V}_{id}(t+1)=\omega\boldsymbol{V}_{id}(t)+c_1r_1(\boldsymbol{P}_{id}-\boldsymbol{X}_{id}(t))+c_2r_2(\boldsymbol{P}_{gd}-\boldsymbol{X}_{id}(t)) \quad (1)$$

$$\boldsymbol{X}_{id}(t+1)=\boldsymbol{X}_{id}(t)+\boldsymbol{V}_{id}(t+1) \quad (2)$$

式中：c_1、c_2——认知因子和社会因子，其值为$c_1=c_2=2$；$r_1,r_2\in[0,1]$是两个服从均匀分布的随机数；t——迭代的代数；$\boldsymbol{P}_{id}$——粒子i搜索到的个体最好位置的第d维分量；$\boldsymbol{P}_{gd}$——整个种群搜索到的全局最好位置的第d维分量；ω——惯性权重，一些

收稿日期：2007-03-06 **E-mail**：fengyuyu@163.com
基金项目：国家自然科学基金项目(60673062)；广东省自然科学基金项目(06025686)；广东省科技计划基金项目(2005B10101048、2006B11201003)。
作者简介：冯玉宇 (1977－)，男 (土家族)，贵州沿河人，硕士研究生，研究方向为进化计算； 周育人 (1965－)，男，湖南岳阳人，博士，副教授，研究方向为演化计算、并行计算。

研究人员[3-4]在大量实验的基础上总结出惯性权重ω从0.9线性递减到0.4效果比较好。

式(1)中的第1部分是惯性部分，代表了粒子对当前运动状态的信任，能够为粒子提供必要的能量；第2部分是认知部分，代表了粒子的思考能力，鼓励粒子飞向自身最好位置；第3部分是社会部分，代表了粒子的交际能力，能够引导粒子飞向种群最好位置。这3个部分的协调和制约关系决定了算法的性能。

1.2 带收缩因子的PSO——PSO-CF

1999年Clerc提出带收缩因子的粒子群算法[5]，他认为带收缩因子的粒子群算法比惯性权重的粒子群算法更简单，且具有良好的收敛性和不用限制最大速度的特点。PSO-CF算法的速度更新方程按式(3)计算

$$V_{id}(t+1)=K*(V_{id}(t)+c_1r_1(P_{id}-X_{id}(t))+c_2r_2(P_{gd}-X_{id}(t))) \quad (3)$$

$$K=2/|2-\varphi-(\varphi^2-4*\varphi)^{1/2}|$$

式中：$\varphi=c_1+c_2$，通常$\varphi=4.1$。

从式(3)可以看出，PSO-CF算法只需要c_1、c_2两个参数，比惯性权重的PSO少一个参数ω。而且，如果不考虑两个随机数r_1和r_2，可以把式(3)改为[6]

$$V_{id}(t+1)=K*(V_{id}(t)+\varphi*(P_{md}-X_{id}(t))) \quad (4)$$

式中：$P_{md}=(c_1*P_{id}+c_2*P_{gd})/(c_1+c_2)$。

从式(4)可以看出，速度更新方程只受认知因子c_1和社会因子c_2之和φ的影响，通常都取$\varphi=4.1$，这样社会因子可以改为$c_2=\varphi-c_1$。

因此，带收缩因子的PSO只需要控制一个参数，其模型比惯性权重的PSO更简单。当$\varphi=4.1$时，$K\approx0.7298$，其值小于1，经过多次迭代后，粒子的速度V将趋近于0，因此PSO-CF算法具有良好的收敛性。

Eberhart和Shi[7]对该算法进一步进行研究，认为限制最大速度$V_{max}=X_{max}$效果更理想(他们在实验中取$c_1=c_2=2.05$)。后来，Carlisle和Dozier[8]对认知因子和社会因子进行研究，认为$c_1=2.8$、$c_2=1.3$效果更好。

详细讨论该算法已超出本文的范围，在实验中，我们取$c_1=2.8$、$c_2=1.3$和限制最大速度$V_{max}=X_{max}$。

1.3 扰乱粒子速度的PSO——PSO-DV

Swagatam Das和Amit Konar等人[9]与差分进化算法结合，提出轻微扰乱粒子速度的算法。其操作是：对粒子i，随机选择另外两个粒子j和k(i≠j≠k)，构造一个新向量δ，按式(5)设置δ

$$\delta=X_k-X_j \quad (5)$$

式中：X_k、X_j——粒子k和j的当前位置。粒子i的第d维速度更新方程如式(6)

$$V_{id}(t+1)=\omega V_{id}(t)+\beta(\delta)_d+c_2r_2(P_{gd}-X_{id}(t)) \quad \text{if rand}(0,1)<CR$$

$$=V_{id}(t) \quad \text{else} \quad (6)$$

式中：CR——交叉率，δ_d——式(5)中定义的向量δ的第d维分量，$\beta\in[0,1]$，文献[9]取$\beta=0.8$，其余参数与惯性权重的PSO算法相同。

粒子i第(t+1)次探测到的探测点Tr_i可表示如下

$$Tr_i=X_i(t)+V_i(t+1) \quad (7)$$

PSO-DV算法与基本粒子群算法及上面介绍的两种粒子群算法更新粒子位置的方法都不一样，它并不是每次都改变粒子的位置，而是在粒子搜索到更好位置时，粒子才会移动到这个更好的位置。

对于求极小值的函数，可如下形式化表示粒子i的位置更新过程

$$X_i(t+1)=Tr_i \quad \text{if } (f(Tr_i)<f(X_i(t))$$

$$X_i(t+1)=X_i(t) \quad \text{else}$$

为了保持粒子的移动性，PSO-DV算法采用重新随机初始化粒子位置的方法，即对粒子i，如果在搜索空间中某点停滞N代(N是能够容忍粒子停滞的最大代数)，那么粒子将被随机初始化一个新位置。

其形式化表示如下：

$$\text{if } ((X_i(t)=X_i(t+1)=\cdots=X_i(t+N)) \text{and } (f(X_i(t+N))\neq f^*)) \text{ then}$$

$$X_i(t+N+1)=X_{min}+\text{rand}(0,1)*(X_{max}-X_{min})$$

这里，f^*是种群搜索到最好点的值，N是能够容忍粒子停滞的最大代数，X_{min}和X_{max}分别表示搜索空间的上、下边界。

从上面的叙述可以看出，PSO-DV算法对惯性权重的PSO算法进行了3点改进：

(1)用一个差分向量扰乱粒子的认知能力；

(2)粒子只能移动到更好的位置；

(3)重新随机初始化粒子位置。

2 扰乱认知能力的带收缩因子的PSO——DPSO-CF

从上面的分析中我们可以看出，PSO-CF算法受收到缩因子的影响，使得粒子以较快的速度聚集到种群已经搜索到的全局最好位置附近，造成了种群失去多样性、粒子早熟收敛的结果。惯性权重线性递减的PSO算法收敛性较差，如果继续对粒子的认知能力进行扰乱，则势必导致粒子更加难以收敛。

受PSO-DV算法的启发，结合PSO-CF算法收敛性好但容易陷入局部极值导致早熟收敛的特点，我们试着改进PSO-CF算法。在改进算法中，我们同样利用一个向量来扰乱粒子的认知能力，但我们所用的向量与PSO-DV算法构造的向量不一样。

PSO-DV算法构造的向量与所要扰乱的粒子之间几乎没有联系，导致粒子丧失认知能力，这种扰乱方法未必是一种好方法，可能引入的扰乱向量不但没有给粒子带来有用的信息，反而影响了粒子自身的“思考”行为，造成粒子漫无目的地“飞行”，导致粒子无法收敛。那么，怎样才能保持粒子的认知能力呢？我们认为保留粒子的个体最好位置是使粒子具有良好认知能力的最好方法，因此，我们构造的向量δ保留了粒子的个体最好位置P_i。

我们构造向量δ的方法是：对群中每个粒子i，都随机选择另一个粒子j(i≠j)，按公式(8)设置δ

$$\delta=P_i-X_j \quad (8)$$

式中：P_i——粒子i的个体最好位置，X_j——粒子j的当前位置。

从式(8)可以看出，构造的向量δ是用粒子i的个体最好位置P_i减粒子j的当前位置X_j，按矢量运算法则，向量δ指向粒子i的个体最好位置P_i，因此，粒子i将向其个体最好位置P_i移动，这样就尽量保持了粒子i的认知能力。

新算法中，我们尽量按PSO-CF算法的速度更新方程更新粒子的速度，只是按一定的小概率用向量$\boldsymbol{\delta}$的某些维替换粒子i相应的维，这种替换扰乱了粒子的“思考”行为，可能会给粒子带来一些有用的信息，帮助粒子逃出局部极值的束缚，避免粒子早熟收敛。对于粒子的位置更新方程，我们采用PSO-CF算法的位置更新方法，而不是采用PSO-DV算法提出的粒子只能移动到更好位置的方法。我们采用的方法可能会使粒子进入到错误的搜索区域，但是粒子的搜索区域更宽广，对增加种群多样性和避免粒子早熟收敛可能会有所帮助。因此，新算法中粒子i的第d维速度和位置更新方程分别按公式(9)和(10)计算。

$$\boldsymbol{V}_{id}(t+1)=\boldsymbol{K}*(\boldsymbol{V}_{id}(t)+c_1r_1\boldsymbol{\delta}_d+c_2r_2(\boldsymbol{P}_{gd}-\boldsymbol{X}_{id}(t))) \quad \text{if rand}(0,1)<CR$$
$$=K*(\boldsymbol{V}_{id}(t)+c_1r_1(\boldsymbol{P}_{id}-\boldsymbol{X}_{id}(t))+c_2r_2(\boldsymbol{P}_{gd}-\boldsymbol{X}_{id}(t))) \text{ else} \quad (9)$$
$$\boldsymbol{X}_{id}(t+1)=\boldsymbol{X}_{id}(t)+\boldsymbol{V}_{id}(t+1) \quad (10)$$

这里，$CR\in(0,1)$是交叉率，$\boldsymbol{\delta}_d$是式(8)中定义的向量$\boldsymbol{\delta}$的第d维分量，其余参数与带收缩因子的PSO算法相同。

PSO算法优点之一就是简单，不需要记忆那么多信息，因此，在新算法中我们没有象PSO-DV算法那样重新随机初始化粒子的位置。

新算法中，我们只是加入扰乱向量$\boldsymbol{\delta}$对PSO-CF算法进行了一点改进，其余操作还是按PSO-CF算法进行，因此，新算法的时间复杂性和空间复杂性与PSO-CF算法几乎一样。

新算法的主要伪代码如下：

```
Procedure DPSO-CF
Begin
initialize population;
while stopping condition isn't satisfied do
    for i = 1 to no_of_particles
    calculate fitness of particle;
    update P_i and P_g ;
    select another particle j (i≠j) randomly;
    construct the vector as δ = P_i-X_j;
    for d = 1 to no_of_dimensions
        if rand(0,1)<CR
        V_id(t+1) = K* (V_id(t)+c1r1δ_d+c2r2 (P_gd−X_id(t)));
        else
        V_id(t+1)=K*(V_id(t)+c1r1 (P_id−X_id(t))+c2r2 (P_gd−X_id(t)));
        endif
    endfor
  endfor
end while
end
```

3 数值实验

为了验证新算法(DPSO-CF)的性能，我们用5个Benchmark函数[9-10]测试新算法，并和上面介绍的PSO-LIR、PSO-CF和PSO-DV这3个PSO算法进行比较。实验中，允许各算法迭代的最大代数为5000，若在设置的最大迭代代数内不能搜索到最优解，则停止迭代；所有Benchmark函数的搜索精度都设置为0.001。每个算法都独立运行50次，通过比较算法成功搜索到最优解的次数和搜索到最优解需要迭代的代数来分析算法的性能。对于PSO-DV算法，我们按文献[9]提供的参数进行设置，种群规模为200、CR=0.9、β=0.8，允许粒子停滞的最大代数N设置为100。

各算法详细的参数设置如表1所示。

表1 PSO算法参数设置

Algorithm	ω	c_1	c_2	Swarm size
PSO-LIR	[0.9,0.4]	2	2	20
PSO-DV	[0.9,0.4]	β=0.8	2	200
PSO-CF	1.0	2.8	1.3	20
DPSO-CF	1.0	2.8	1.3	20

选用的5个Benchmark函数各自具有不同的特点，能够在一定程度上检验算法的优化能力和优化速度。表2列出了函数的维数、极小值/极值点、初始化位置/速度范围、搜索空间/速度范围。

f_1是连续的单峰函数，用于检验算法收敛速度；f_2是一个经典的复杂优化函数，取值区间内走势平坦，用于测试算法收敛速度；$f_3\sim f_5$是复杂的非线性多峰函数，具有大量局部极值，用于检验算法的全局搜索能力和逃离局部极值能力。

表3是新算法与其它3个PSO算法实验结果对比；表4是新算法取不同扰乱率时实验结果对比。

说明：S-Times表示成功搜索到最优解的次数；G_{avg}表示搜索到最优解平均迭代代数。

CR是DPSO-CF算法的扰乱率。

从表3可以看出，新算法搜索到最优解的次数明显要多于其它3个PSO算法，这说明新算法具有良好的稳健性。新算法通过加入扰乱向量对带收缩因子的PSO算法进行改进，改进算法搜索到最优解的次数明显比PSO-CF算法要多，这说明这个扰乱向量能够帮助粒子逃离局部极值的束缚，避免粒子早熟收敛；同时，改进算法搜索到最优解需要迭代的平均代数比PSO-CF算法要多，这就说明增加扰乱向量后，粒子的搜索区域更宽广了，所以也就增加了种群的多样性。新算法搜索到最优解需要迭代的代数比惯性权重线性递减的PSO算法

表2 Benchmark函数及参数设置

名称	函数	维数	极小值(极值点)	初始化位置/速度范围	搜索空间/速度范围
Hyper-ellipsoid	$f_1(x)=\sum i*x^2$	20	$0(x_i=0)$	$(-5.12,5.12)^n$	$(-5.12,5.12)^n$
Rosenbrock	$f_2(x)=\sum[100*(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	20	$0(x_i=1)$	$(-50,50)^n$	$(-50,50)^n$
Rastrigin	$f_3(x)=nA+\sum[x_i^2-A\cos(2\pi x_i)]$ A=1	20	$0(x_i=0)$	$(-5.12,5.12)^n$	$(-5.12,5.12)^n$
Ackley	$f_4(x)=e+20-20*\exp^{-0.2*(1/n\Sigma x_i^2)^{1/2}}-\exp^{1/n\Sigma\cos(2\pi x_i)}$	20	$0(x_i=0)$	$(-32,32)^n$	$(-32,32)^n$
Griewank	$f_5(x)=1/4000\sum x_i^2-\prod\cos(x_i/i^{1/2})+1$	20	$0(x_i=0)$	$(-600,600)^n$	$(-600,600)^n$

表3 DPSO-CF与其它3个PSO算法实验结果对比

Func	PSO-LIR		PSO-DV		PSO-CF		DPSO-CF		
	S-Times	G_{avg}	S-Times	G_{avg}	S-Times	G_{avg}	CR	S-Times	G_{avg}
f_1	22	2628	0	>5000	28	235	**0.02**	**44**	**322**
f_2	15	3528	0	>5000	1	576	**0.02**	**47**	**869**
f_3	42	2839	0	>5000	0	>5000	**0.04**	**47**	**1245**
f_4	27	3056	0	>5000	2	369	**0.04**	**45**	**1583**
f_5	8	2864	0	>5000	5	337	**0.02**	**17**	**1035**

表4 DPSO-CF取不同扰乱率时实验结果对比

Func	0.005		0.010		0.020		0.040		≥0.060	
	S-Times	G_{avg}	S-Times	G_{avg}	S-Times	G_{avg}	S-Times	G_{avg}	S-Times	G_{avg}
f_1	37	245	41	257	**44**	**322**	42	815	0	>5000
f_2	5	608	30	748	**47**	**869**	45	2790	0	>5000
f_3	2	286	2	391	2	417	**47**	**1245**	0	>5000
f_4	20	490	33	451	44	611	**45**	**1583**	0	>5000
f_5	9	319	14	462	**17**	**1035**	3	3917	0	>5000

要少得多，这说明新算法具有良好的收敛性。

从表4可以看出，对于不同的函数，需要取不同的扰乱率才能使新算法达到最佳效果，这就提出一个问题，选择怎样的扰乱率才合适呢。当CR≥0.06时，新算法搜索不到最优解，我们认为J.Kennedy在文献[11]中对粒子群算法的解释能够说明这个问题：粒子间高度的社会联系并不比粒子间适度的社会联系更能找到最优解。

4 结束语

受PSO-DV算法的启发，结合PSO-CF算法收敛性好的特点，本文提出扰乱粒子认知能力的PSO-CF算法，实验结果表明，这种轻微的扰乱是有益的，能够在一定程度上增加种群多样性，帮助粒子逃出局部极值的束缚，避免粒子早熟收敛。新算法搜索到最优解需要迭代的次数比带收缩因子的PSO算法要多，但比惯性权重线性递减的PSO算法要少，这说明新算法具有良好的收敛性。

同时，我们也看到：为了使新算法得到比较理想的结果，针对不同的函数，需要取不同的扰乱率，这就提出如何选择扰乱率的问题，它是否与种群规模、函数的维数及震荡性有关系呢，这些都还需要我们进一步进行探索。

参考文献：

[1] Shi Y,Eberhart R C.Fuzzy adaptive particle swarm optimization [C].Proc Congress on Evolutionary Computation,2001.

[2] Carlisle A, Dozier G. Adapting particle swarm optimization to dynamic environments[C].Las Vegas Nevada USA:Proceedings of the International Conference on Artificial Intelligence,2000: 429-434.

[3] Jaco F Schutte,Albert A Groenwold.A study of global optimization using particle swarms[J]. Journal of Global Optimization, 2005,31:93-108.

[4] Shi Y,Eberhart R C.Empirical study of particle swarm optimization[C].Angeline P J,Michalewicz Z,Schoenauer M,et al.Proceedings of the Congress of Evolutionary Computation.Washington DC,USA:IEEE Press,1999:1945-1950.

[5] Clerc M.The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization[C].Angeline P J,Michalewicz Z,Schoenauer M,et al.Proceedings of the Congress of Evolutionary Computation. Washington DC, USA: IEEE Press, 1999: 1951-1957.

[6] Rui Mendes,James Kennedy,José Neves.The full informed particle swarm: simpler, maybe better[J].IEEE Transaction on Evolutionary Computation,2004,8(3):204-210.

[7] Eberhart R C,Shi Y.Comparing inertia weights and constriction factors in particle swarm optimization[C]. Proceedings of the 2000 Congress on Evolutionary Computation. Piscataway, NJ: IEEE Service Center,2000:84-88.

[8] Carlisle A,Dozier G.An off-the-shelf PSO[C]. Indianapolis, USA: Proceedings of the Workshop on Particle Swarm Optimization, Purdue School of Engineering and Technology,2001.

[9] Swagatam Das,Amit Konar,Uday K Chakraborty.Improving particle swarm optimization with dierentially perturbed velocity[C]. Washington DC,USA: Genetic and Evolutionary Computation Conference(GECCO),2005.

[10] Liang JJ,Qin AK.Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J].IEEE Transaction on Evolutionary Computation,2006,10(3):281-296.

[11] Kennedy J.Small-worlds and mega-minds: Effects of neighborhood topology on particle[C]. Washington DC: Pro 1999 Conf Evolutionary Computation,1999:71-76.